New ways for building BCK-algebras of higher order

Radu Vasile*

Ovidius University, Bd. Mamaia 124, Constanţa, 900527, România
E-mail: rvasile@gmail.com

Abstract

In this paper we present new extensions for BCK-algebras. Also, we give a representation of the BCK-algebras class structure, by introducing a new concept, BCK-trees.

2010 Mathematics Subject Classification. **06F35** 18B35
Keywords. BCK-algebras, BCK-algebra extension, BCK-tree.

1 Introduction

BCK-algebras were first introduced in 1966 by Y. Imai and K. Iseki, in [4], as a generalization of the difference operation of two sets and generalization of propositional calculi. For example, the relation

$$A - (A - B) \subset B, A, B \text{ sets},$$

in propositional calculi can be written

$$p \rightarrow ((p \rightarrow q) \rightarrow q).$$

Recently, there were revealed new interesting applications of these new algebras, such as those found in [5] and [3]. In the first of the two articles, a "block code" is constructed from a BCK-algebra. At the end of the article, the authors ask if the reverse procedure is possible. In the second quoted article, a partial answer is given to this question, showing that, under certain conditions, the answer is affirmative, it is possible to construct a BCK-algebra starting from a "block code". Surprisingly, however, the BCK-algebra obtained from the code is not isomorphic with the BCK-algebra from which the same code was obtained, which means that code can have 2 associated BCK-algebras. In the article [3] it is shown that these algebras are in an equivalence relation called "code similar". Moreover, it is shown that the two algebras do not even have the same properties, one being commutative and non-implicative, the other being non-commutative and non-implicative. On a closer look, it can be noticed that in the development of the codes from BCK-algebras, or, of the BCK-algebras from codes, what is actually used is the table of operations that the algebra generates. The definition of BCK-algebra gives us the uniqueness of the codewords, obtained by applying some transformations on the operations table of that algebra. Since what counts in getting the codewords is the operations table, we can have isomorphic algebras, with different codes attached. Hence, it is important to study all BCK-algebras, not just the equivalence class representatives from the isomorphism relation. For this, it is important to see how we can construct BCK-algebras directly, by computations, or by the help of the computer and to isolate various sets of BCK-algebras, based on their properties. In the process of constructing such an algebra, we want to obtain an algebra with certain properties, or, when we talk about the construction of new BCK-algebras, by extension from some existing ones, we want not to lose any properties.

*Special big thanks to conf. univ. dr. Cristina Flaut, Ovidius University, Constanţa, România

Tbilisi Mathematical Journal Special Issue (IECMSA-2019), pp. 111–124.
Tbilisi Centre for Mathematical Sciences.

Received by the editors: 15 March 2020.
Accepted for publication: 24 April 2020.

2 Preliminaries

2.1 BCK-algebras

Definition 2.1 ([6, 7])**.** A set $(X, *, 0)$ of type $(2, 0)$ is called a *BCI*-algebra if the following conditions are satisfied:

1. $((x * y) * (x * z)) * (z * y) = 0$, $\forall x, y, z \in X$;
2. $(x * (x * y)) * y = 0$, $x, y \in X$;
3. $x * x = 0$, $\forall x \in X$;
4. If $x * y = 0$ and $y * x = 0$, then $x = y, \forall x, y \in X$.

Definition 2.2 ([6, 7])**.** If a BCI-algebra X satisfies the following relations:

5. $0 * x = 0$, $\forall x \in X$,

then X is called a *BCK*-algebra.

In a BCI-algebra, the following relations also apply [7, pp 6] and [6]:

6. $(x * y) * z = (x * z) * y$;
7. $(x * y) * x = 0$;
8. $((x * z) * (y * z)) * (x * y) = 0$;
9. $x * 0 = x$;

Some authors give an alternative BCI-algebra definition, replacing axioms from the original definition with one of those listed above, proving that it is also a well-defined axiomatic system for BCI-algebra [see [2] prop. 3.1]:

Proposition 2.3. A BCI-algebra is characterized by the following axioms:
1, 4, 9,
and a BCK-algebra through the axioms:
1, 4, 9, 5.

Definition 2.4 ([3])**.** If $(X, *, 0)$ and $(Y, \circ, 0)$ are two BCK-algebras, an application $f : X \to Y$ with the property $f(x * y) = f(x) \circ f(y)$, for any $x, y \in X$, is called a BCK-algebras morphism. If f is a bijective application, then f is a *isomorphism* of BCK-algebras.

Remark 2.5. It can be observed that we can define a partial order relation $<$ on a BCK-algebra by the formula $x < y$ if and only if $x * y = 0$.

2.2 Properties of BCI / BCK-algebras

Definition 2.6 ([6])**.** In a BCI / BCK-algebra X if there is a generic element denoted with 1 that satisfies the relation $x < 1, \forall x \in X$, element 1 is called the unit of X algebra. A BCI / BCK-algebra with unit is called a bounded algebra.

Definition 2.7. [6] In a bounded BCK-algebra, an element with the property $1 * (1 * x) = x$ is called an involution.

Definition 2.8. A BCI / BCK-algebra X is called *commutative* if $x * (x * y) = y * (y * x)$, any $x, y \in X$.

Definition 2.9. A BCI / BCK-algebra x is called *implicative* if $x * (y * x) = x$, any $x, y \in X$.

Definition 2.10. A BCI / BCK-algebra $X, *, 0$ is called positive-implicative, if it satisfies the relation: $(x * z) * (y * z) = (x * y) * z$, for all x , $y, z \in X$.

Remark 2.11 ([2])**.** A BCI-Commutative Algebra is a commutative BCK-Algebra.

Remark 2.12. A BCI-algebra is implicative if it is commutative and positive-implicative.

In [6], the Iseki extension of a BCK-algebra is presented. This extension is a way to obtain new higher-order BCK-algebras from lower-order ones and is defined as follows:

Definition 2.13. Let B be a BCK-algebra. This can be extended to a Bounded BCK-algebra by adding a 1 element that complies with the following rules:

1. $1 * 1 = x * 1 = 0, \forall x \in B$.
2. $1 * x = 1, \forall x \in B$.

Remark 2.14. If a BCK-algebra B is a positive-implicative BCK-algebra, then the Iseki extension is a positive-implicative BCK-algebra.

Remark 2.15. If a BCK-algebra B is commutative, the Iseki Extension is not commutative. Considering that it does not preserve commutativity, the extension of Iseki does not preserve the property of being implicative.

3 Main results

3.1 New types of extensions for BCK-algebras

In the previous section, it is seen that BCK-algebras can be constructed by extension from the order n to the order $n + 1$ through the Iseki extension. This is defined by introducing a new element and extending the operation from the source algebra to the resulting algebra. In the following, we build new extensions for BCK-algebras.

3.1.1 The Pseudo-Iseki Extension

Proposition 3.1. Let $(X, *, 0)$ be a BCK-algebra of order n and an element d that does not belong to X with the following properties:

1. $d * d = 0$;

2. $0 * d = 0$;
3. $d * x = d, \forall x \in X$;
4. $x * d = x, \forall x \in X$;

Then the algebra $X[d]$ is a BCK-algebra.

Proof. To be a BCK-algebra, the newly constructed algebra must satisfy the system of axioms. From the way we chose the element d, we have $d * d = 0$, $0 * d = 0$ and $d * x = d$, $x * d = x$, whatever $x \in X$. Thus, the axioms 3, 4, 5 in the axiomatic system of BCK-algebras are fulfilled. For axiom 2 we have $(d * (d * y)) * y = 0, \forall y \in X$ because $d * y = d$ and $d * d = 0$. On the other hand $(x * (x * d)) * d = 0, \forall x \in X$ because $x * d = x$ and $x * x = 0$.
We still have axiom 1 of the axiomatic system. $((d * y) * (d * z)) * (z * y) = 0$ because the first two brackets give d, and $d * d = 0$.
$((x * d) * (x * z)) * (z * d) = 0$ because $x * d = x$ and $z * d = z$, and the expression is reduced to: $(x * (x * z)) * z$ which is 0 from the definition of BCK-algebra X.
$((d * y) * (d * d)) * (d * y) = 0$ because the middle bracket is canceled, and the extreme brackets are similar. $((x * y) * (x * d)) * (d * y) = 0$ because: $x * d = x$, $d * y = d$, so the expression is reduced to $((x * y) * x) * d$. By the definition of the element d, the expression is reduced to $(x * y) * x$, this being 0 according to relation 7 in Definition 2.2. Q.E.D.

Definition 3.2. The extension from the previous proposition will be called pseudo-Iseki extension.

Proposition 3.3. If the source algebra is commutative, by pseudo-Iseki extension a commutative algebra can be obtained. If the source algebra is a positive-implicative one, the algebra obtained by pseudo-Iseki extension is a positive-implicative one.

Proof. Let $(X, *, 0)$ be a commutative and positive-implicative BCK-algebra. For commutativity, we have $x*(x*d) = d*(d*x)$. $(d*x) = d$, so $d*(d*x)) = 0$. $(x*d) = x$, so $x*(x*d)) = 0$. On the other hand, we have $d*(d*y) = y*(y*d)$. $(d*y) = d$, so $d*(d*y) = 0$. $(y*d) = y$, so, $y*(y*d) = 0$. The case where all the elements are d is trivial, since $d * d = 0$, and $d * 0 = d$, the final relation being $d = d$. For positive-implicative we have the following expressions: $(d*z)*(y*z) = (d*y)*z$. $(d * z) = d$, whence the expression becomes $d * (y * z) = (d * y) * z$. $(d * y) = d$, and $d * z = d$. so, the expression is $d * (y * z) = d$, which is true from the definition of the element d.
$(x*d)*(y*d) = (x*y)*d$. The parentheses from the left hand expression are x respectively y, and the right hand expression is $x * y$, so the entire expression is true. Last expression to be checked: $(x * z) * (d * z) = (x * d) * z$. $(d * z) = d$, $(x * d) = x$ and the expression becomes $(x * z) * d = x * z$ which is true from the definition of the element d . Q.E.D.

Remark 3.4. Regarding the order relation defined on the BCK-algebra, the pseudo-Iseki extension introduces a new element that does not compare with any of the elements in the source algebra.

3.1.2 The involution extension

In the following, we propose an interesting new extension. We propose it for research because it cannot be applied to all BCK-algebras, but only to certain algebras, thus defining a interesting type of BCK-algebras. Based on this extension, we can derive a construction process, applied to the operation tables of some BCK-algebras, to obtain higher-order BCK-algebras. Interesting to this process is that it does not affect the commutativity property and in some cases, it does not affect the positive-implicative property either.

Proposition 3.5. Let $(X, *, 0)$ be a BCK-algebra of order n and $d \notin X$. Respecting the following relations, we can obtain a BCK-algebra of order $n+1$:

1. $x * d = 0$ any $x \in X$;
2. $d * d = 0$;
3. $d * x = y$ if and only if $d * y = x$ any $x, y \in X$;
4. $((d * y) * (d * z)) * (z * y) = 0$ any $x, y, z \in X$.

Proof. From the choice of the element d, the axioms 3, 4, 5 of the axiomatic system of BCK-algebras are respected. For the axiom 1, regardless of the expression from 4, the only problematic relation remains: $((x*y)*(x*d))*(d*y) = 0$. But $(x*d) = 0$, so the expression reduces at $(x*y)*(d*y) = 0$. The element $(d * y) \in X)$ and $x = (d * z)$, $z \in X$, hence we can write: $((d * z) * y) * (d * y) = 0$. Applying a transformation we obtain: $((d * z) * (d * y)) * y = 0$. This expression should be true for any $y, z \in X$, therefore we can exchange y with z and the relation becomes: $((d*y)*(d*z))*z = 0$. In the relation 4 we have $((d * y) * (d * z)) * (z * y) = 0$. This means that $(d * y) * (d * z) < (z * y)$, but $z * y < z$, whence $(d * y) * (d * z) < z$ this mean $((d * y) * (d * z)) * z = 0$. Axiom 2 remains to be verified. For the 2 axiom we have the cases:

1. $(d * (d * y)) * y$. In this case $d * y$ can be x with the property as $d * x = y$, or $d * y = y$. In the first case we have $d * y = x$, and $d * x = y$ where $y * y = 0$. In the second case, $d * y = y$, and finally $y * y = 0$.
2. $(x * (x * d)) * d$. Here $x * d = 0$, $x * 0 = x$, and finally $x* = 0$.

Q.E.D.

Remark 3.6. We notice that in the previous extension, 0 is placed on the column of the element d, and on the line of the element d there are only elements with the property of involution, or auto involution, that is: $d * (d * x) = x$, respectively $d * x = x$.

Definition 3.7. We will call the extension defined previously the involution extension of a BCK-algebra of order n to an algebra of order $n + 1$.

Proposition 3.8. If a BCK-algebra $(X, *, 0)$ is commutative, the BCK-algebra resulting from the involution extension is also a commutative BCK-algebra.

Proof. We have the following cases:

1. $d * (d * y) = y * (y * d)$. In this case $y * d = 0$, $y * 0 = y$. $d * (d * y) = y$.
2. $x * (x * d) = d * (d * x)$. In this case, $x * d = 0$, and $x * 0 = x$. $d * (d * x)) = x$.

Q.E.D.

Proposition 3.9. Let $(X, *, 0)$ be a positive-implicative BCK-algebra. The BCK-algebra of order $n+1$ obtained by the extension through involutions is positive-implicative, if $(d*z)*(y*z) = (d*y)*z$.

Proof. We have the following cases:

1. $(d * z) * (y * z) = (d * y) * z$. It is true from the statement.
2. $(x * z) * (d * z) = (x * d) * z$. The right hand expression is 0. So it remains to be shown that $(x * z) * (d * z) = 0$. This expression is 0 from the definition of extension by involution, see the proof for axiom 1.
3. $(x * d) * (y * d) = (x * y) * d$. True, because the parentheses in front of the equals are 0, and after the equals, the expression is 0 from the definition of the element d.
4. $(x * d) * (d * d) = (x * d) * d$. The expression before equals is 0 from the definition of element d, and the one after equals is also 0.
5. $(d * d) * (y * d) = (d * y) * d$. The first parenthesis is 0, the entire expression becomes 0, and after all, everything is 0, from the definition of the element d.

Q.E.D.

Remark 3.10. Such a BCK-algebra exists:

*	0	1	2	3
0	0	0	0	0
1	1	0	1	0
2	2	2	0	0
3	3	2	1	0

Remark 3.11. In 3.9, we have the condition $(d*z)*(y*z) = (d*y)*z)$ which should be fulfilled by the resulting BCK-algebra to be a positive-implicative one. If we take $(d * y) = z)$ and $(d * z = y)$, we notice that the expression transforms in $y * (y * z) = z * z$, hence we have $y * (y * z) = 0$. In conclusion, if we have $(X, *, 0)$a positive-implicative BCK-algebra, an element $d \notin X$, satisfing the properties from the involution extension definition, for each pair of elements $(x, y) \in X$ with the property that $(d * x) = y$ and $d * y = x$, we must have $x * (x * y) = y * (y * x) = 0$

Remark 3.12. The following BCK-algebra does not obey the above rule:

*	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	2	2	0	0
3	3	2	1	0

$1 * (1 * 2) = 1$, $2 * (2 * 1) = 0$.
If we take the relation from the proposition: $(3 * 2) * (1 * 2) = (3 * 1) * 2$. $(3 * 1) = 2$, $(1 * 2) = 0$, $3 * 2 = 1$, so te expression becomes $1 * 0 = 2 * 2$, which means $1 = 0$ false.

Remark 3.13. By this process we can obtain bounded commutative BCK-algebras, starting from commutative BCK-algebras.

Remark 3.14. When passing from the order n, with n an even number, to the order $n+1$ at least one element on the line of the element d is a self-involution, that is $d * x = x$.

Proposition 3.15. Let $(X, *, 0)$ be a positive-implicative BCK-algebra of order n, with n an even number. If we want to extend this BCK-algebra to a BCK-algebra of order $(n + 1)$, using the involution extension, it must be a bounded BCK-algebra.

Proof. In 3.14 we have noticed that going from n, an even number, to $(n+1)$ trough involution extension, gives us a self-involution element on the last row, let it be y. It has the property that $(d*y) = y$. From the proof of the involution extension, we have seen that the following relation should be fulfilled: $(x*y)*(d*y) = 0$, for all $x, y \in X$. But there is an y with the property $d*y = y$, so, for this element, the expression reduces to $(x*y)*y = 0$. In any positive-implicative BCK-algebra, we have the relation $(x*y)*y = (x*y)$ [see [6]]. Therefore $(x*y) = 0$ for y with the property $d*y = y$. Hence the source algebra should be bounded with y the greatest element.

Q.E.D.

Remark 3.16. As we see, the involution extension does not play nicely with
the positive-implicative property. In some cases, it makes this property to vanish, in other cases, we can not even apply this extension to positive-implicative BCK-algebras. On the other hand, it plays very nicely with the commutative property.

We further define a construction process, which is a particular case of this extension.

Corollary 3.17. Suppose we have a BCK-algebra X of order n, whose elements are numbered in the order of their appearance in the operations table from top-to down, 0 to $n-1$. We introduce a new element that we note with n, which respects the properties of the element d from the definition of extension by involution, with the following specific configuration of the table:

1. On the column of element n, all elements are 0;
2. On the line of the element n, all the elements are placed in reverse order from the one in the table. That is, the line looks like this: $n, (n-1), (n-2), ..., (n-(n-2)), 1, 0$

Remark 3.18. From the construction, we notice that the elements are involutions, that is: $n*0 = n$, and $n*n = 0$, $(n*1 = (n-1)$, and $n*(n-1) = 1$. In the case of passing from an order n even, to an order $n+1$ odd, the element in the middle of the line will be an auto involution IE: $n*([(n/2)]+1) = [(n/2)]+1)$.

Remark 3.19. It can be observed in some examples, that BCK-algebras obtained by extension through involutions, are isomorphic with BCK-algebras obtained by the process presented in 3.17. For example, we take the following two BCK-algebras:

*	0	1	2	3	4
0	0	0	0	0	0
1	1	0	0	0	0
2	2	2	0	3	0
3	3	3	0	0	0
4	4	2	1	3	0

*	0	1	2	3	4
0	0	0	0	0	0
1	1	0	0	0	0
2	2	2	0	0	0
3	3	3	2	0	0
4	4	3	2	1	0

The two algebras are isomorphic by changing 3 to 2, through an isomorphism, leaving the other elements in place. It remains to be studied how we can extend this example to a general result.

Remark 3.20. It also remains to be studied if there are ways to detect those BCK-algebras to which this extension can be applied, using only elements or properties of the source algebra, without involving the d element. Thus, given a BCK-algebra of order n, we can say whether or not this extension can be applied without using the element d.

3.2 Structure of the BCK-algebras class

In the following, we give some interesting results about the structure of the class of BCK-algebras and the importance of this structure. We introduce the notion of BCK-trees and their properties. First of all, we should pay close attention to how BCK-algebras can be build directly.

3.2.1 Construction of BCK-algebras

The axiomatic system generates some constraints on the operation table as follows:

1. On the 0 element line, there are only 0 values because $0 * x = 0$; from axiom 5;
2. Diagonally we only have 0 according to $x * x = 0$; from axiom 3
3. On the column of the 0 element, each element has its own value because $x * 0 = x$; from relation 9
4. We can not have 0 at the same time on position (i, j) and on position (j, i) with $j \neq i$ because $x * y = 0, y * x = 0 \to x = y$; from axiom 4
5. If the algebra is bounded, in the 1 column, the values are 0, because $x * 1 = 0$.

Basically, starting from these simple rules, you can build BCK-algebras and calculate the number of algebras of a particular order. So we find that we have a single algebra of order 2 that is bounded and implicative.

Example 3.21. Let X = 0, a.

$*$	0	a
0	0	0
a	a	0

For order 3, applying the axioms we get the following skeleton:

$*$	0	a	b
0	0	0	0
a	a	0	x
b	b	x	0

where we noted with x missing elements that must be completed. Note that the simplest rules at a 3 per 3 table leaves only two available positions, with a constraint however, we cannot put 0 on both positions where we have x. We obtain the following possible algebras:

Alg 1.

$*$	0	a	b
0	0	0	0
a	a	0	0
b	b	a	0

Alg 2.

$*$	0	a	b
0	0	0	0
a	a	0	0
b	b	b	0

Alg 3.

$*$	0	a	b
0	0	0	0
a	a	0	a
b	b	0	0

Alg 4.

$*$	0	a	b
0	0	0	0
a	a	0	b
b	b	0	0

Alg 5.

$*$	0	a	b
0	0	0	0
a	a	0	a
b	b	a	0

Alg 6.

$*$	0	a	b
0	0	0	0
a	a	0	b
b	b	b	0

Alg 7.

$*$	0	a	b
0	0	0	0
a	a	0	a
b	b	b	0

Alg 8.

$*$	0	a	b
0	0	0	0
a	a	0	b
b	b	a	0

Considering that we used the axioms 3, 4, 5 and 9 in the construction of the table and using the result in 2.3, we must check only the axiom 1. By calculations, the first 4 algebras are BCK-algebras, isomorphic two by two, 1 with 4 and 2 with 3. The algebra in 7 is a BCK-algebra, the rest being neither BCK-algebra nor BCI. We have 3 BCK-algebras of the order 3, unique to an isomorphism. For the study, we will retain as algebra representations 1, which is bounded, commutative but not positive-implicative, 2, bounded, positive-implicative but non-commutative and 7 implicative, therefore positive-implicative and commutative.

3.2.2 Structure of the operation table of a BCK-algebra

In 3.2.1 we have seen that the axiomatic system places certain constraints on the way the table of operations of a BCK-algebra looks. Starting from these constraints, we can construct BCK-algebra templates, templates that must then be filled in with the missing elements. For the order 3, the template looks like this:

$*$	0	a	b
0	0	0	0
A	A	0	x
b	b	x	0

Assuming we have an algebra of order 3 and we want to obtain, by extension, an algebra of order 4, we have the following template:

$*$	0	a	b	c
0	0	0	0	0
A	A	0	x	xx
b	b	x	0	xx
C	C	xx	xx	0

where, through x, we noted the unknown elements of the order 3, and through xx we noted the new elements introduced by the extension to the order 4. Such a template can be extended to the order n. We notice that if we have an algebra of order $n-1$ and we want to extend it to the order n, the axiomatic system leaves us, on the last column, $n-2$ elements to complete, and on the last line we have also $n-2$ elements, because $x*0=x$, $x*x=0$, $0*x=0$. In other words, the axiomatic system of BCK-algebras always fixes the corners of the algebra operating table. Each of the $n-2$ positions left on the last line and the last column can be filled with elements from 0 to $n-1$, that is n possibilities. By computation, the total number of possibilities for combining elements is $n^{(n-2)}$ for the line and $n^{(n-2)}$ for the column. This leads to a total number of possible combinations $(n^{(n-2)})^2$. From this total number, we must subtract, however, the constraints placed by the axiom according to which $x*y=0$ and $y*x=0 \to x=y$. In other words, in the template we build, we cannot put 0 at the same time on the corresponding positions, so if $x*y=0$, then the $y*x$ must be different from 0. The situations given by this axiom are $\sum_{i=1}^{n-2} C_i^{n-2} = 2^{(n-2)}-1$. Through these calculations, we demonstrated the following result:

Proposition 3.22. The total number of tables among which we can find a BCK-algebra when making the transition from the order $n-1$ to the order n, is $(n^{(n-2)})^2-(2^{(n-2)}-1)$.

By this result, we can calculate, for a certain order n, what is the number of tables among which we can find BCK-algebras, considering that the previous formula applies to any order.

In [6] Theorem 2.6 it is shown that any BCK-algebra of order n contains a BCK algebra of order $n-1$. We can therefore always see a BCK algebra of order n, as an extension of a BCK algebra of order $n-1$, obtained by bordering the operations table with a line and with a column. We get:

Proposition 3.23. The total number of n order templates to be checked when identifying n order BCK-algebras is:
$\prod_{i=2}^{n}(i^{(i-2)})^2-(2^{(i-2)}-1)$.

With this result, we observe that from very small orders it is difficult to find BCK-algebras, both by direct calculation and by the process of order-to-order extensions. Over this previously calculated number, if we consider an algorithm for verifying the axioms of BCK-algebras, we observe that for each such generated table, we have to perform n^3 operations to verify that the generated table is a BCK-algebra. So a legitimate question can be asked: Can we get a better algorithm for generating BCK-algebras? Can we somehow get BCK-algebras with certain properties in a faster way?

3.2.3 BCK-trees

In [6], Theorem 2.6 it is shown that any BCK-algebra of order n contains a BCK algebra of order $n-1$. Moreover, it is stated that a commutative algebra contains only commutative algebras. This perspective on BCK-algebras has always been seen from the top-down direction, from larger algebras to smaller algebras. By changing the perspective in reverse and looking from the bottom up, we can present the following conclusions, contained in the following results.

Proposition 3.24. Any BCK-algebra of order n, $n \geq 3$ contains at least one subalgebra of order 3

Proof. According to [6], theorem 2.6, every BCK-algebra of order n contains a BCK algebra of order $n-1$. Let $(X, *, 0)$ be a BCK-algebra of order n and $Y, *, 0)$, $Y \subset X$ a subalgebra in X of order $n-1$. Then Y contains a subalgebra of order $n-2$. Therefore, a chain of subalgebras is formed that goes down to the order of 3. Q.E.D.

Proposition 3.25. If a BCK-algebra contains a non-commutative subalgebra, it is non-commutative.

Proof. It emerges immediately from the definition of commutativity, given that it must be valid for any two elements x, y that belong to algebra. Q.E.D.

In fact, the proposition regarding commutativity is valid for any of the properties of BCK-algebras that refer to all elements of algebra. Thus, we can say:

Proposition 3.26. If a BCK-algebra contains a subalgebra that is not positive-implicative, it is not positive-implicative.

Corollary 3.27. If a BCK-algebra of order n contains an algebra of order 3 which is not commutative or positive-implicative, then it is not commutative or positive-implicative.

Proof. It emerges immediately from the propositions 3.24, 3.25, 3.26 Q.E.D.

In [6], Theorem 2.7, the exact number of subalgebras of a given order that a BCK-algebra of order n has is calculated. Thus, we can see that a BCK-algebra of order n contains many BCK-algebras of order 3.

Remark 3.28. Using the bottom-up perspective, we see that we can construct higher-order BCK-algebras starting from 3 algebra. And if the 3 algebra from which we start doesn't have a certain property, any higher-order BCK-algebra will not have that property.

Thus, starting from the BCK-algebra from 2, only higher non-commutative algebras can be obtained, because this is non-commutative. On the other hand, starting from this algebra, we are likely to find positive-implicative BCK-algebras. Starting from the BCK-algebra from 1, only higher-order algebras that are not positive-implicative can be obtained, because this is not positive-implicative, but is possible to find commutative algebras. Noting that the algebras in 1 and 2 are not implicative, it follows that any higher-order algebra chain starting from them will be non-implicative. The algebra from 7 is implicative, so commutative and positive-implicative. If we start from this, we have chances to obtain higher-order, implicative BCK algebras. Given that we have 3 algebras of order 3, unique to an isomorphism, it follows that we have 3 possible starting points on which to build higher-order BCK-algebras. Thus, true BCK-algebra trees are created, starting from the 3 types of BCK-algebras of the order of 3, trees which we will call BCK-trees.

Definition 3.29. The set of BCK-algebras, obtained by successive extensions from a lower order to a higher order, starting from a BCK-algebra of order 3 isomorphic with one of the algebras from 1, 2, 7, is called a BCK-tree.

Remark 3.30. BCK-trees are not independent, they intersect where, a higher-order algebra contains BCK-algebras of order 3 isomorphic with 2 or even all 3 types of algebra of order 3 from 1, 2 and 7, forming a true BCK-algebra graph. At these intersection points of the BCK-trees, the higher-order algebras, resulting from the extension, lose properties. They take the set of common properties held by the 3 order algebras they contain.

Remark 3.31. The intersection of the BCK-trees is made by means of BCK-algebras of the same order which are isomorphic between them.

For example, if we take the algebra from 7 and apply the Iseki extension, we get the following algebra:

$*$	0	1	2	3
0	0	0	0	0
1	1	0	1	0
2	2	2	0	0
3	3	3	3	0

By applying the Iseki extension, the resulting 4 order BCK-algebra contains an 3 order BCK-algebra isomorphic with the algebra of 2, losing the commutativity. Therefore, this algebra should be found in the BCK-tree generated by 7, the one we started from, but also in the tree generated by 2, the one with which it intersected. Given these things, we should be able, starting from the BCK-algebra of 2, to obtain the same BCK-algebra of the order of 4, or one isomorphic with it. Using the computer, we generated all BCK-algebras of order ≤ 5, and on the order of 4, starting from the algebra of 2, we found the following algebra:

$*$	0	1	2	3
0	0	0	0	0
1	1	0	0	1
2	2	2	0	2
3	3	3	0	0

This is obviously isomorphic with the previous algebra, through the application that switches the elements 2 and 3 between them, leaving in place the elements 0 and 1. We can now issue an important observation as a conclusion of this subsection:

Remark 3.32. The commutativity and positive-implicative properties of a BCK-algebra n with $n \geq 4$ depend on the properties of the BCK-algebras of order 3 that it contains as subalgebras.

3.2.4 The importance of BCK-trees

Regarding this new structure, we observe a similarity with the Cayley Dickson process from algebras built over a field, a process by which, at each step of transition from one dimension to another, certain properties are lost. This structure is useful to be studied, because it shows us where we can go to get algebras with certain properties. As we have seen, starting from one of the 3 order BCK-algebra types, a certain property may not exist. The importance of this structure is even greater if we consider the equivalence relations between the BCK-algebras class and other logical algebra classes. We know that there are links between Hilbert algebras and positive-implicative

BCK-algebras, [see [1]], between MV-algebras and bounded commutative BCK-algebras [see [8]], as well as between Boolean algebras and implicative bounded BCK-algebras [see [6]].

In this sense, if we want to find Boolean algebras, among the BCK-algebras, they will have to be searched only by extensions of the BCK-algebra of order 3 from the 7, because this is the only BCK-tree that contains implicative bounded BCK-algebras. Knowing that Boolean algebras have the order 2^n, we can find Boole algebras among BCK-algebras in the associated tree of the algebra from 7, taking the algebras which have the order 2^n. Specifically, we can generate them using an extension from a BCK-algebra of the order $2^n - 1$, positive-implicative and commutative, from that tree, using such an extension that gives us a bounded algebra and does not destroy the commutativity and positive-implicative property. The only extension to generate bounded BCK-algebras known is the Iseki extension, but, as we have seen, it destroys commutativity so it cannot be used for this purpose. In this paper, we proposed the involution extension as a mechanism to obtain bounded commutative BCK-algebras and, in some circumstances, bounded implicative BCK-algebras.

Regarding MV-algebras, given the equivalence relation with bounded and commutative BCK-algebras, it is clear that they will not be searched in the tree associated with the 2 algebra, because this algebra is not commutative. So we have the other two trees. We have noticed that in the tree associated with the algebra in 7 we have implicative algebras, and some of these can be bounded. Moreover, we have shown, by equivalence with Boolean algebras, that there exist algebras having the order 2^n in this tree that are bounded and implicative. In fact, we get a stronger result:

Proposition 3.33. Any bounded BCK-algebra with other order than 2^n, is not implicative.

However, bounded BCK-algebras, in this tree, are also found at other orders than 2^n, and even there are algebras of order 2^n that are not implicative. Indeed, it is sufficient to apply the Iseki extension to a BCK-algebra of order $2^n - 1$ in this tree, to obtain such a bounded non-implicative BCK-algebra of order 2^n. The bottom line is that for these bounded algebras with a different order than 2^n, one of the properties disappears, either the commutativity or the positive-implicative property. This means that they are at a point of intersection of this tree generated by the algebra of 7 with one of the other two trees. If the intersection is made with the tree generated by the algebra in the 2, the commutativity is lost, so this path does not help us. We can therefore conclude that MV-algebras are found either in the tree generated by the algebra in 1, or at the points of intersection of this tree with the tree generated by the algebra in 7, the algebra in 1 being itself a commutative bounded BCK-algebra. Using this assumptions and the fact that Boolean algebras class is a proper subclass of MV-algebras class, we can derive the following observation:

Remark 3.34. The difference between MV-algebras and Boolean Algebras in therm of BCK-algebras Properties is the positive-implicative property. This property is lost by MV algebras.

Finally, regarding the equivalence relation between Hilbert algebras and positive-implicative BCK-algebras, we can find these algebras in the trees generated by the algebras of 2 and 7, these being trees that contain positive-implicative algebras.

4 Conclusions

In this paper, we presented two new ways for extending BCK-algebras and the effect of these new extensions on BCK-algebras properties. They should be further investigated, to see how they are related with other properties, like the property to be with the condition(s.). In particular, The Involution extension could be further investigated to define exactly the class of BCK-algebras to which it can be applied directly, without checking the outside extension element. Also, we structured the class of BCK-algebras by introducing the BCK-tree concept. BCK-trees separate subclasses of BCK-algebras, based on their properties, so we have the tree of commutative BCK-algebras, the tree of positive-implicative BCK-algebras and the tree of Implicative BCK-algebras. This structure shows also the relation between BCK-algebras and other classes of logical algebras. Moreover, based on this, we build some conclusions about implicative BCK-algebras, conclusions that can be more developed in future research.

References

[1] W.A. Dudek, *On embedding Hilbert algebras in BCK-algebras*, Mathematica Moravica **3** (1999) 25–28.

[2] W.A. Dudek, *Remarks on the axioms system for BCI-algebras*, *Prace Naukowe WSP w Czestochowie* **2** (1996) 46–61.

[3] C. Flaut, *BCK-algebras arising from block-codes*, Journal of Intelligent & Fuzzy Systems **28** (2015) 1829–1833.

[4] Yasuyuki Imai and Kiyoshi Iseki, *On axiom systems of propositional calculi, XIV*,Proceedings of the Japan Academy **42** (1966).

[5] Y. B. Jun and S. Z. Song, *Codes based on BCK-algebras*, Inform. Sciences. **181** (2011) 5102–5109.

[6] J. Meng and Y. B. Jun, *BCK-Algebras*, Kyung Moon Sa Co., Seoul, (1994).

[7] S. R. Mukkamala, *A Course in BE-algebras*, Springer Nature Singapore Pte Ltd., Singapore, (2018).

[8] C. Flaut and R. Vasile, *Wajsberg algebras arising from binary block-codes*, Soft computing **24** (2020) 6047–6058.